REMARKS

The claims remaining in the present application are Claims 16-32. The instant specification has been amended. The drawing has been amended. No new matter has been added as a result of these amendments.

OBJECTIONS

The drawings are objected to. A replacement sheet and an annotated sheet showing the changes made to the drawings has been submitted herewith. No new matter has been added as a result of these amendments. Applicants respectfully request removal of the objection to the drawings.

The specification is objected to. The specification has been amended to correct an informality. No new matter has been added as a result of this amendment. Applicants respectfully request removal of the objection to the specification.

REJECTIONS

Claims 16-32 are rejected under 35 U.S.C. §102(b) as being anticipated by Kelly et al., U.S. Patent No. 5,832,205 (hereinafter, Kelly). The rejection is respectfully traversed for the following reasons. Claims 16-32 are neither taught nor suggested by Kelly.

Claim 16 recites:

> A method of determining validity of a translated instruction comprising:
> a) starting execution of a first host instruction translated from a first target instruction, wherein said first host instruction is linked from a second host instruction translated from a second target instruction, and

> wherein a first condition of a target system state required by said first host instruction holds;
>
> b) testing a second condition of said target system state to determine the validity of said first host instruction;
>
> c) executing said first host instruction if said second condition holds; and
>
> d) generating an exception if said second condition does not hold.

Claim 16 recites that a second condition of the target system state is tested to determine the validity of a first host instruction. Applicants respectfully submit that Kelly fails to teach or suggest this claimed limitation.

The rejection cites the "T-bit" as meeting this claimed limitation. However, Applicants do not understand the "T-bit" to be used to determine the validity of a first host instruction, as claimed. Rather, the T-bit is used to mark pages for which valid translations exist, such that target memory is not overwritten. Thus, the T-bit is not used to determine validity of a host instruction as claimed, but rather is used to prevent target memory from being overwritten.

> The translation bit (T-bit) is used to indicate target memory pages for which translations exist. The T-bit thus possibly indicates that particular pages of target memory contain target instructions for which host translations exist. If an attempt is made to write to the protected pages in memory, the presence of the T-bit will cause an exception which when handled by the code morphing software can cause the appropriate translation(s) to be invalidated or removed from the translation buffer. The T-bit can also be used to mark other target pages that translation may rely upon not being written. (Col. 22, lines 35-48, emphasis added).
>
> Prior to executing the translation, the T bit for the target page(s) containing the target instructions that have been translated is set. This indication warns that the instruction has been translated; and, if an attempt to write to the target address occurs, the attempt generates an exception which causes the translation to possibly be invalidated or removed. (Col. 23, lines 29-35, emphasis added).

Determining that a write to a given target address could lead to overwriting a target instruction for which a valid translation exists, does not provide the necessary information to determine the validity of a host instruction, as claimed. That is, if the testing of the T-bit indicates that there is no attempt to write to a target page, then no exception occurs and no information at all is garnered about the validity of a given host instruction. If no information is gained about a host instruction, then the Applicants respectfully assert that testing the T-bit cannot constitute, "testing a second condition of said target system state to determine the validity of said first host instruction," as claimed.

If the testing of the T-bit indicates that there is an attempt to write to a target page, then an exception occurs. The exception may cause a translated instruction to be invalidated. However, the fact that a translation may be removed from the translation buffer when processing the exception does not meet the claimed limitation of "testing a second condition of said target system state to determine the validity of said first host instruction." The claimed limitation of testing to determine the validity implies a test that determines that the instruction is *either valid or invalid*. Applicants note that c) of Claim 1 recites, "executing said first host *instruction if said second condition holds*." Thus, a possible outcome of the testing of the second condition must be that the second condition holds or, in other words, that the first instruction is valid. Testing of the T-bit *cannot determine that a host instruction is valid*, according to Applicants understanding of Kelly. Therefore, testing of the T-bit is not a test of the validity or invalidity of a host instruction, as claimed. Consequently, Applicants do not understand Kelly to teach or suggest using the T-bit to determine the validity of a host instruction, as claimed.

For the foregoing rationale, Claim 16 is neither taught nor suggested by Kelly. As such, allowance of Claim 16 is respectfully requested.

Claim 24 recites:

> A method of determining validity of a translated instruction comprising:
>
> a) performing a first address consistency check of a first host instruction made from a first target instruction to verify that said first host instruction is valid;
>
> b) executing said first host instruction;
>
> c) determining whether a second host instruction made from a second target instruction and that is linked from said first host instruction can be safely executed without a second address consistency check; and
>
> d) executing said second host instruction without performing said second address consistency check if safe.

Claim 24 recites "performing a first address consistency check of a first host instruction made from a first target instruction to verify that said first host instruction is valid." For at least the reasons provided in the response to Claim 16, this claimed limitation of Claim 24 is neither taught nor suggested by Kelly.

Furthermore, Claim 24 recites, "determining whether a second host instruction made from a second target instruction and that is linked from said first host instruction can be safely executed without a second address consistency check." The rejection cites Kelly col. 17, lines 1-39 as meeting this claimed limitation. Applicants respectfully submit that this passage fails to teach or suggest this claimed limitation. In order to maintain a rejection under 35 U.S.C. §102, all limitations of the claim must be taught in the prior art reference, arraigned as in the claim.

> Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim (*Lindemann Maschinefabrik GmbH* v. *American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984)).

In contrast, to meeting the claimed limitation, the cited passage is concerned with the operation of the gated store buffer, which controls the transfer of data to memory. The passage indicates that memory stores generated during execution of host instructions are placed in the store buffer. Upon the occurrence of a commit operation, the memory stores generated during the execution are moved past the gate (committed) and written to memory. However, Applicants do not understand this passage to teach or suggest the claimed limitation of "determining whether a second host instruction made from a second target instruction and that is linked from said first host instruction can be safely executed without a second address consistency check." For the foregoing rationale, Claim 24 is neither taught nor suggested by Kelly. As such, allowance of Claim 24 is respectfully requested.

Claim 29 recites:

> A method of linking translated instructions comprising:
> a) translating a first target instruction to a first host instruction;

b) determining that said first host instruction is to be linked to a second host instruction translated from a second target instruction; and

c) providing an address consistency check for said first host instruction.

As Applicants have previously noted, anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, <u>arranged as in the claim</u>. The rejection cites the code sequence in columns 25 and 26 of Kelly as teaching the limitations of Claim 29. Applicants first argue that *the cited code in columns 25 and 26* in Kelly fails to teach or suggest "determining that said first host instruction is *to be linked* to a second host instruction translated from a second target instruction," as claimed. Because *the cited code in columns 25 and 26* in Kelly does not teach this claimed limitation, this code example does not present each and every element of the claimed invention, <u>arranged as in the claim</u>. Thus, the cited code in columns 25 and 26 does not support a rejection under 35 U.S.C. §102.

The rejection appears to consider the claimed limitation of "determining that said first host instruction is to be linked to a second host instruction translated from a second target instruction" to be taught by two instructions following one another in a sequence in the code example in column 25. That is, Applicants understand the rejection to interpret "linked instructions" as any two instructions that follow one another. Applicants submit that one of ordinary skill in the art, reading the claims in light of the specification, would not interpret the cited code in column 25 as "determining that instructions are to be linked," as claimed. The instant specification teaches that linking is described in Kelly (page 2, lines 8-9 of instant specification). Kelly describes linking from col. 18, line 53 – col. 20, line 24, and in Figure 8. Applicants highlight the following passage in Kelly.

> Since the second set of target instructions follows the first set of target instructions, the primitive branch instruction at the end of the host translation of the first set of target instructions is automatically updated to substitute the address of the host translation of the second set of target instructions as the branch address for the particular condition controlling the branch. (Kelly col. 19, lines 36-42.)

Figure 8 illustrates the linking of instructions as indicated by the "update address" notation linking the two sets of instructions. One of ordinary skill in the art would understand that the translation process being described with respect to the code example at column 25 of Kelly is not the linking process that Kelly describes from col. 18, line 53 – col. 20, line 24, and in Figure 8. Thus, the code example at column 25 does not teach or suggest "determining that said first host instruction is to be linked to a second host instruction translated from a second target instruction," as claimed.

Applicants further argue that the Claim 29 limitation of "providing an address consistency check for said first host instruction," overcomes the cited reference. The Applicants have claimed that the address consistency check is for the first host instruction. The rejection cites the "chkl" and "chku" instructions in the example in col. 26 of Kelly as teaching this claimed limitation. However, the "chkl" and "chku" instructions are not used as address consistency checks for a host instruction, as claimed. That is, it is not the address of a host instruction that is being checked, but rather the address of data to be accessed by a host instruction. Referring to the code example in col. 26, first the host "add" instruction loads the address of data to be accessed into a register (host register R0). Then the "chkl" instruction determines if the *address of the data* is within range of a segment. However, the "chkl" instruction does not perform an address consistency check of the host instruction itself, as claimed.

For the foregoing rationale, Claim 29 is neither taught nor suggested by Kelly. As such, allowance of Claim 29 is respectfully requested.

Claims 17-23, 25-28, and 30-32 depend from Claims 16, 24, and 29, which are believed to be allowable for the foregoing rationale. As such, Claims 17-23, 25-28, and 30-32 are believed to be allowable.

For the following additional rationale, Claim 17 is patentable over Kelly. Claim 17 recites:

> The method of Claim 16, wherein said first condition is based on an <u>address consistency</u> check of said second host instruction (emphasis added).

Claim 17 recites that the first condition is based on an address consistency check of a host instruction. The rejection cites the A/N bit taught by Kelly as meeting this claimed limitation. Applicants respectfully submit that Kelly fails to teach or suggest an <u>address consistency</u> check by using the A/N bit. The A/N bit is not used to test the consistency of an address, but is rather used to test whether an <u>access type</u> matches the A/N protection. The A/N bit may be set in the translation look-aside buffer to indicate either a memory page or memory-mapped I/O, that is, the type of memory access.

> As the access is attempted, the target memory reference is checked by comparing the access type presumed (normal or abnormal) against the A/N protection bit now in the TLB page table entry. When the access type does not match the A/N protection, an exception occurs. If the operation in fact affects memory, then the optimizing, reordering, and rescheduling techniques described above were correctly applied during translation. If the comparison with the A/N bit in the TLB shows that the operation, however, affects an I/O device, then execution causes an exception to be taken; and the translator produces a new translation one target instruction at a time without optimizing, reordering, or rescheduling of any sort. Similarly, if a translation incorrectly assumes an I/O operation for an operation which actually affects memory, execution causes an exception to be taken; and the target instructions are retranslated using the optimizing, reordering, and rescheduling techniques. (Col. 28, lines 22-45. emphasis added).

The above passage from Kelly clearly indicates that the A/N bit is used to compare an access type with a protection type. However, the A/N bit is not used to determine a consistency between addresses, as claimed. Therefore, Claim 17 is patentable over Kelly.

For the additional foregoing rationale, Claim 17 is neither taught nor suggested by Kelly. As such, allowance of Claim 17 is respectfully requested.

CONCLUSION

In light of the above listed amendments and remarks, reconsideration of the rejected Claims is requested. Based on the amendments and arguments presented above, it is respectfully submitted that Claims 16-32 overcome the rejections of record and, therefore, allowance of Claims 16-32 is earnestly solicited.

Should the Examiner have a question regarding the instant amendment and response, the Applicants invite the Examiner to contact the Applicants' undersigned representative at the below listed telephone number.

Respectfully submitted,
WAGNER, MURABITO & HAO LLP

Dated: 11/25, 2003

Ronald M Pomerenke
Ronald M. Pomerenke
Registration No. 43,009

Address:
WAGNER, MURABITO & HAO LLP
Two North Market Street
Third Floor
San Jose, California 95113

Telephone:
(408) 938-9060 Voice
(408) 938-9069 FAX